ISRN SICS-T--91/4--SE

A Bibliography on Sketches from the Computer Science Point of View

by
Oskar Permvall

SICS technical report
T91:04
ISSN 1100-3154

A Bibliography on Sketches from the Computer Science Point of View

Oskar Permvall
Department of Computer Science
Lund University
E-mail: oskar@dna.lu.se

May 1990

SICS Report T91–04

Abstract

A sketch is a categorical tool for presenting theories. This is a commented bibliography on sketches and their use within some areas of computer science. References to work on sketches not primarily related to computer science are also provided.

1 Introduction

As category theory is becoming more and more used in computer science there is a need for bibliographies on areas of research where category theory and computer science meet. This one collects references on sketches and related topics from category theory and computer science. The comments to the bibliography are divided in subsections depending on the connection with different areas of computer science. Entries are ordered alphabetically by author(s).

I am making this bibliography available as an aid and service to those interested in sketches. I appreciate any and all help you can give towards correcting any mistakes that may be present and in providing additional references.

2 Sketches

A *sketch* is a categorical tool for presenting various kinds of theories. These theories can also be named categorical theories and as such be considered as a development of the work of F. W. Lawvere on algebraic theories [Law63].

Formally, a basic sketch is (see [BarWel85, Gra87]) a four-tuple $(\mathcal{G}, U, D, C)$ where $\mathcal{G}$ is a directed graph, U is a function assigning a special arrow to each node of the graph $\mathcal{G}$, D is a class of diagrams in $\mathcal{G}$ and C is a class of cones in $\mathcal{G}$. Diagrams and cones are defined as in ordinary category theory and in a model of a sketch they should be mapped to commutative diagrams and limit cones.

There is a hierarchy of sketches based on properties of the components of the four-tuple, model properties, and some extensions. The lower levels of this hierarchy are roughly equivalent to data types and programs[1].

For the mathematically inclined the basic reference is the book by M. Barr and C. Wells [BarWel85] where the hierarchy is described in connection with topos and triple theory. Their new book [BarWel90] contains a good introduction to the lower part of the sketch hierarchy and its connection to computer science. The more advanced uses of sketches are only touched upon and the reader is referred to their previous book [BarWel85] for a treatment, but then there are no examples from computer science. Barr and Wells motivate this partitioning by 'These generalizations are described without much detail since they do not appear to have many applications (yet!) in computer science'. Barr and Wells point out that their new book can be seen as an introduction to their previous book [BarWel85].

B. Pierce has written a report on category theory for computer scientists in which he also evaluates several books and papers in this area [Pie88]. His opinion about a draft of the new book by Barr and Wells is 'an excellent addition to the literature' but also 'this book is not for the faint-hearted'.

Sketches were discovered by Charles Ehresmann in the late 1960's, though in a form slightly different from the one above, see section 2.7 below for a discussion of that version.

Theories have always been of great interest in both logic and computer science. Barr and Wells are working with abstract theories (no syntax, rules, or method of generation is specified) and a sketch is a presentation of such a theory. A related but more concrete approach is taken by J. Goguen and R. Burstall in two papers about algebraic tools for the semantics of computation [GogBur84b, GogBur84c]. Their approach is that presentations are signatures with equations and consequently their theories are of a syntactic nature. These theories are put to work in languages such as Clear [BurGog81, BurGog80] and the OBJ family [GogWin88] designed by them and their colleagues.

M. Fourman and S. Vickers have written a short but comprehensive paper about theories, categories, and their relations [FouVic86]. In the proceedings where that paper appeared are also some introductory papers by A. Poigné on categories and various topics from logic and algebra.

2.1 Sketches and the Theory of Datatypes

The usefulness of category theory in this area is undisputable and the ADJ group (J. Goguen, J. Thatcher, E. Wagner and J. Wright) made a good initial work to pave the way for further developments, see [Gog+78] for examples and further references.

Sketches appear to have larger potential in describing various aspects of data types, such as subsorts, conditional equations and the like. Ordinary textual specifications rely on some kind of syntax and some special constructions to handle these

[1] Models of a finite product sketch in a suitable category is equivalent to the algebras of a data type specified with an equational signature. The category of models of an equational Horn theory is sketchable by a finite limit sketch [Bar89]. Horn clause logic is essential for the formalisation of logic programming [Pad88].

aspects, where sketches have it all in a coherent framework of objects and arrows without using variables. The reason for this belief is that certain types of sketches come equipped with both cones and cocones which can express constraints that have no counterpart within initial semantics unless one extends the formalism with some new concepts. Various approaches have been tried, e.g. order sorted algebra [Gog+85, GogMes87] and the languages ACT ONE and ACT TWO where the latter has a logic for specifying constraints [Fey88, Cla89, EhrMah89]. It must, however, be said that these approaches focus on different topics; Goguen and his colleagues on rewriting, unification etc. while the ACT-group focuses on modules and their interconnections.

An introduction to sketches and data types is the paper by Wells and Barr [WelBar88]. Another useful introduction is their new book [BarWel90] where category theory and sketches are explained in a computer science context. J. Gray has developed a theory of parameterisation of data types based on sketches [Gra87]. It is worth mentioning that one of his main sources of inspiration was [Tha+82] which is a development and an extension of the initial semantics in [Gog+78].

Types, type theory, polymorphism and their interrelations are a very active area of research where sketches can be useful. A basic reference is [Bai+90]. Gray has treated the relation between these topics and sketches in three papers [Gra88, Gra89a, Gra89b]. In connection with this there is also a further development of his algebraic data type theory.

Among other approaches to a categorical treatment of data types are the *Categorical Programming Language* by T. Hagino [Hag87a, Hag87b] which is further studied by G. Wraith [Wra89]. One should note that on page 22 in [Hag87a] Hagino mentions that it would be interesting to investigate his CSL by means of sketches. A different but related approach is presented by H.-D. Ehrich [Ehr82b].

Data types are to be used within some kind of programming language. The use of sketches in the context of functional languages is discussed in [Wel90] and [BarWel90] as mentioned above.

Equational logic is too limited to describe all aspects of data types and so are sketches for some programming language structures. To remedy this Wells [Wel90] has developed an extension to sketches called *form* which can be used to specify entities other than limits and colimits in a model, e.g. function space objects and objects such as lists which are defined by recursion.

Of related interest is Wagner's treatment of imperative languages [Wag90]. Since sketches are a specification method, free from variables based on composition of operations it seems to fit functional languages better than imperative languages but it is difficult to say in which direction research will move.

2.2 Sketches and Knowledge Bases

The first paper to appreciate the usefulness of sketches in this area was [Mat85, Mat86]. In developing sketches whose models are to be used as knowledge bases one needs to use such types of sketches as coherent and geometric which are at the upper level of the hierarchy and whose models reside in various kinds of toposes.

Hopefully this approach to knowledge representation will be developed in such a way that it can be useful for default reasoning and also be a fruitful approach to non-monotonic reasoning in artificial intelligence [Lin90].

Using categories to model knowledge bases is also done by a group of researchers at the Department of Mathematics, Instituto Superior Técnico in Lisbon, Portugal, but instead of sketches they use institutions as a framework for knowledge theories [Fia+88, Ser+90]. Another interesting approach to the model theory of knowledge is M. A. Nait Abdallah's theory of ions, which is an extension of logic programming [Nai89].

Databases, both knowledge and ordinary, play a central role in computer science and instances of such can be viewed as models of suitable theories, precisely as data types can be seen as models (algebras) of equational theories, for a general discussion, see [Mak84].

Naturally, it is possible to use sketches to model databases since they are simpler than knowledge bases. There exist other categorical models of databases and in [Per90] we will undertake a comparison of them and a sketch model of a database.

2.3 Sketches and Institutions

The theory of institutions was presented and developed in several papers by Goguen, Burstall and others starting with [GogBur84a]. The aim of this theory, which resembles abstract model theory [Bar74], is to be able to formalise and abstract the relation between syntax and semantics of various logics.

In the last section of [Gra87] Gray develops a connection between sketches and institutions via fibred categories. Given the concept of model of a sketch, it determines a functor from the category of sketches to the category of small categories. Via this functor we obtain a fibred category over the category of sketches. This fibred category is a simple example of an institution.

The fibred category in connection with a category of structured sketches[2] gives a generalisation of the main theory developed in [Gra87]; moreover, it would be nice if one could use sketches as a foundation for the theory of institutions.

2.4 Sketches, Algorithms and Computation

The development of sketches shortly presented below in section 2.7 is more related to various areas of mathematics than to computer science, with one exception, the work of R. Guitart. He has taken the theory of sketches as a foundation for a 'geometry of computations' which is presented in two papers [Gui86b, Gui88]. In these papers he describes sketches of programs, homotopies of theories, how to sketch the finiteness of models and other topics associated with a geometry of computations. This work seems to have started with his earlier work on algebraic analysis [Gui82b, Gui86a, Gui87], figurative algebras [Gui82a] and other notions, all of which are also related to sketches.

[2]A structured sketch is a way of parametrising a sketch with another sketch under certain requirements expressed by a subsketch. The constructions discussed by Gray focuses on the class of **Obj**-structured sketches.

2.5 Sketches and Neural Networks

As we have emphasised, sketches are a powerful categorical tool to study knowledge bases and theories. An instance of a knowledge base is a model of the sketch presenting the theory governing the knowledge base. This model is to be implemented in one way or another.

In the last ten years there has been a renewed interest in neural networks and their use as computing devices. There is hope that neural networks could serve as implementations of models of sketches presenting knowledge theories. The main reason for this hope is the work of L. Shastri [Sha88, Sha89] in which a connectionist realisation of a memory network based on evidential reasoning is described.

Shastri concentrates his investigation on questions about inheritance, recognition and efficiency of computation. Translating sketches into his formalism or vice versa gives an opportunity to use his realisation as a model of a suitable sketch. Neural networks as models seem to be only a few steps further away.

Two central ideas in knowledge theories are distributed and local representation. These ideas are in a sense inherent in the work of Shastri and the connection to sketch theory is via sheaf and topos theory which can be used to represent local phenomena.

2.6 Sketches, Categorical Logic and Topos Theory

The connection between logic and category theory has been of great interest for many years. This is a rich field with many variations and complements like theories and models, finitary and infinitary logic, intuitionistic and classical logic, and so on. In connection with computer science it is worth mentioning the correspondence between λ-calculus and cartesian closed categories. This correspondence has implications for higher-order logic [LamSco86].

Two good introductions can be found in *Handbook of Mathematical Logic* [Fou77, KocRey77]. The main topic of the paper by A. Kock and G. E. Reyes is how to make the model of a theory into a functor while M. Fourman concentrates on viewing toposes as theories.

The relation between categories, the world of logical formulas and concepts, and model theory have been described in two expository papers by G. E. Reyes [Rey74, Rey77] where sheafs corresponds to concepts (formulas), sites to theories and topos theory to model theory.

M. Makkai and G. E. Reyes have written a book which is the main reference to this area [MakRey77]. The book clarifies the relation between logical theories and categorical ones such as those treated in [BarWel85], i.e. theories presented by sketches.

M. Barr has contributed to this area with two papers on sketches and their models [Bar86, Bar89]. They are extensions of the work in [BarWel85]. Barr also discusses the relevance of sketches for computer science, especially data types and logic programming, see [Mak87, Mak84, Pad88] for related approaches.

When it comes to computation it looks like Horn logic is of greater importance than first order logic, see for example [GogMes86]. Horn theories taken categorically are treated in a paper by O. Keane [Kea75].

A. Blass has written a quite interesting paper [Bla88] where he argues that the geometric perspective on toposes is closely connected to computation and logic. This seems to be related to topics in the papers by Guitart mentioned above.

Sketches are closely connected to topos theory. Theories of coherent and geometric sketches are toposes, in the latter case it is indeed a Grothendieck topos. The basic reference is the book by Barr and Wells [BarWel85].

Other references includes the standard work on topos theory [Joh77] and the interesting books by J. Bell, R. Goldblatt, J. Lambek and P. Scott [Bel89, Gol84, LamSco86] which all make a connection between logic and the internal language of a topos.

Model theory is an essential part of logic and with a model is often meant a set-valued interpretation of a logical theory. In view of the established connection between logic and category theory it is natural to study models of a theory in any suitable category, e.g. a topos.

A new concept 'categorical model theory' has been coined by M. Makkai and R. Paré and it means the study of set-valued models of possibly infinitary first order theories by means of the conceptual tools of category theory. They develop this theory in the framework of sketches and define a notion of 'accessible category' by which they mean that such a category is sketchable, i.e. equivalent to the category of models of a small sketch. This work has been presented in a paper [Par89] and a recently published monograph [MakPar89].

2.7 Other References on Sketches

As pointed out above there exists an earlier version of sketches, or to state it correctly: Sketches were discovered by Charles Ehresmann in the late 1960's [Ehr68]. A difference is that these sketches are based on categories and not on graphs as is the version defined by Barr and Wells.

Barr and Wells [BarWel85] note and acknowledge that their notion of sketch is a simpler version than that developed by Ehresmann. However, Guitart [Gui86b] and Barr and Wells note that these notions are equivalent, i.e. a translation between them is possible.

The theory and applications of these sketches have been developed in several papers by various authors [Bar71, Bur70b, BasEhr72, BasEhr74, Bur70a, Cop82, CopLai85, CopLaiXX, Die77, EhrEhr78a, EhrEhr78b, EhrEhr79, Fol+80, FolLai72, GuiLai80, GuiLai81a, GuiLai81b, GuiLai82a, GuiLai82b, Gui82c, Gui86b, Gui88, Kel82b, Kel82a, Kel82c, Lai70, Lai75, Lai74, Lai77, Lai79a, Lai79b, Lai87a, Lai87b, Mac84, Mou84, MakPar89, Par89].

The aim and use of Ehresmann's sketches seem to be a bit different and not related to computer science, except from some papers by Guitart which were discussed above in section 2.4.

A summary of the first years of research on sketches can be found in [Bas73]. The complete works of Charles Ehresmann on sketches can be found in [Ehr82a]. Sketches are one of the many mathematical structures that Charles Ehresmann discovered, which is pointed out and discussed by S. MacLane in [Mac80].

2.7.1 Availability of Material

When searching for references to make this bibliography as complete as possible, I found, and this is also other authors' experience, that the journals *Esquisses Mathématiques* and *Diagrammes* have not reached outside France to any great extent.

Here follows information for the benefit of people interested in sketches. There seems to have been published 20 volumes of *Diagrammes* since 1979 and it is available at[3]:

Université Paris VII
U. E. R. de Mathématiques
Tour 45-55, 5$^{\text{ème}}$ Etage
2, place Jussieu
75221 Paris Cédex 05
France

For information about *Esquisses Mathématiques* and the collected works of Charles Ehresmann, see advertisements that appear now and then in the journal *Cahiers de Topologie et Géométrie Différentielle Catégoriques.*

3 Conclusion

The use of sketches in computer science appears to be very promising. Sketches will constitute a good framework for several theories in computer science as will category theory itself, or to state it with the words of Gray in [Gra88],

> As was the case earlier in mathematics, many different formalisms may be proposed for what is essentially the same concept, and new ad hoc structures are erected to support these formalisms. The field is open for category theory to play the same role in these developments that it played earlier in mathematics; namely, clarify concepts, formulate uniform definitions, simplify through abstraction and guide computations. But this time, one doesn't have to create category theory at the same time, because category theory is already in place, replete with its whole arsenal of well understood, ready-made structures.

Acknowledgments

I would like to thank my supervisor Philippe Mathieu for introducing me to the interesting field of sketches. He and Bengt Nilsson have also provided invaluable help when I had problems with the French language. I also wish to thank Olle Lindeberg for many fruitful discussions.

[3]The address is taken from *Diagrammes* vol 13.

References

[Bai+90] E. S. Bainbridge, P. J. Freyd, A. Scedrov, and P. J. Scott. Functorial polymorphism. *Theoretical Computer Science*, 70(1):35–64, 1990.

[Bar71] J.-P. Barthelemy. Esquisses pointées. *Esquisses Mathématiques*, 12, 1971.

[Bar74] K. Jon Barwise. Axioms for abstract model theory. *Annals of Mathematical Logic*, 7:221–265, 1974.

[Bar77] J. Barwise, editor. *Handbook of Mathematical Logic*, volume 90 of *Studies in Logic and the Foundations of Mathematics*. North Holland Publishing Company (Elsevier Science), 1977. (*Reprinted 1989*).

[Bar86] Michael Barr. Models of sketches. *Cahiers de Topologie et Géométrie Différentielle*, XXVII(2):93–107, 1986.

[Bar89] Michael Barr. Models of Horn theories. In Gray and Scedrov [GraSce89], pages 1–7.

[BarWel85] Michael Barr and Charles Wells. *Toposes, Triples and Theories*, volume 278 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag, 1985.

[BarWel90] Michael Barr and Charles Wells. *Category Theory for Computing Science*. Prentice-Hall, 1990.

[Bas73] Andrée Bastiani. Sketched structures and completions. *Cahiers de Topologie et Géométrie Différentielle*, XIV(2):158–160, 1973.

[BasEhr72] A. Bastiani and C. Ehresmann. Categories of sketched structures. *Cahiers de Topologie et Géométrie Différentielle*, XIII(2):104–214, 1972.

[BasEhr74] A. Bastiani and C. Ehresmann. Multiple functors I. *Cahiers de Topologie et Géométrie Différentielle*, XV(3):215–292, 1974.

[Bel89] J. L. Bell.
Toposes and Local Set Theories, volume 14 of *Oxford Logic Guides*.
Oxford University Press, 1989.

[Bla88] Andreas Blass.
Topoi and computation.
EATCS Bulletin, 36:57–65, October 1988.

[Bur70a] A. Burroni.
Esquisses des catégories à limites et des quasi-topologies.
Esquisses Mathématiques, 5, 1970.

[Bur70b] E. Burroni.
Catégories discrètement structurées. Triples.
Esquisses Mathématiques, 4, 1970.

[BurGog80] R. M. Burstall and J. A. Goguen.
The semantics of Clear, a specification language.
In Dines Bjørner, editor, *Proocedings of the 1979 Copenhagen Winter School on Abstract Sofware Specification*, volume 86 of *Lecture Notes in Computer Science*, pages 292–332. Springer-Verlag, 1980.

[BurGog81] R. M. Burstall and J. A. Goguen.
An informal introduction to specifications using Clear.
In Robert Boyer and J. Moore, editors, *The Correctness Problem in Computer Science*, pages 185–213. Academic Press, Inc, 1981.
Reprinted in [GehMcG85], pp 363–390.

[Cla89] I. Claßen.
Revised ACT ONE: Categorical constructions for an algebraic specification language.
In Ehrig et al. [Ehr+89], pages 124–141.

[Cop82] L. Coppey.
Sur quelques structures de base pour définir les structures.
Diagrammes, 7:C1–C23, 1982.

[CopLai85] L. Coppey and C. Lair.
Algébricité, monadicité, esquissabilité et non-algébricité.
Diagrammes, 13:1–111, 1985.

[CopLaiXX] L. Coppey and C. Lair.
Leçons de théorie des esquisses 1–6.
Université Paris VII, UER de Mathématiques.

[Die77] Y. Diers.
Catégories Localisables.
PhD thesis, Université de Paris, 1977.

[Ehr68] Charles Ehresmann.
Esquisses et types de structures algébriques.
Bul. Inst. Politec. Iaşi, 14:1–14, 1968.

[Ehr82a] Andrée Charles Ehresmann.
Charles Ehresmann: Œuvres Complètes et Commentées, Part IV: Esquisses. Complétions. Enrichissements.
Amiens, 1982.

[Ehr82b] H.-D. Ehrich.
On the theory of specification, implementation, and parametrization of abstract data types.
Journal of the ACM, 29(1):206–227, 1982.

[EhrEhr78a] A. C. Ehresmann and C. Ehresmann.
Multiple functors II.
Cahiers de Topologie et Géométrie Différentielle, XIX(3):295–333, 1978.

[EhrEhr78b] A. C. Ehresmann and C. Ehresmann.
Multiple functors III.
Cahiers de Topologie et Géométrie Différentielle, XIX(4):387–443, 1978.

[EhrEhr79] A. C. Ehresmann and C. Ehresmann.
Multiple functors IV.
Cahiers de Topologie et Géométrie Différentielle, XX(1):59–104, 1979.

[EhrMah89] H. Ehrig and B. Mahr.
Fundamentals of Algebraic Specification 2, volume 21 of *EATCS Monographs on Theoretical Computer Science.*
Springer-Verlag, 1989.

[Ehr+89] H. Ehrig et al., editors.
Categorical Methods in Computer Science, volume 393 of *Lecture Notes in Computer Science.* Springer-Verlag, 1989.

[Fey88] Werner Fey.
Pragmatics, concepts, syntax, semantics and correctness notions of ACT TWO: An algebraic module specification and interconnection language.
Technical Report 88-26, TU Berlin, FB 20, 1988.

[Fia+88] José Fiadeiro, Amílcar Sernadas, and Cristina Sernadas.
Knowledgebases as structured theories.
In K. V. Nori and S. Kumar, editors, *Foundations of Software Technology and Theoretical Computer Science, 8th Conference*, volume 338 of *Lecture Notes in Computer Science*, pages 469–486. Springer-Verlag, 1988.

[FolLai72] F. Foltz and C. Lair.
Fermeture standard des catégories algébriques.
Cahiers de Topologie et Géométrie Différentielle, XIII(3):275–307, 1972.

[Fol+80] F. Foltz, C. Lair, and G. M. Kelly.
Algebraic categories with few monoidal biclosed structures or none.
Journal of Pure and Applied Algebra, 17:171–177, 1980.

[Fou77] Michael P. Fourman.
The logic of topoi.
In Barwise [Bar77], pages 1053–1090.

[FouVic86] M. P. Fourman and S. Vickers.
Theories as categories.
In David H. Pitt et al., editors, *Category Theory and Computer Programming 1985*, volume 240 of *Lecture Notes in Computer Science*, pages 434–448. Springer-Verlag, 1986.

[GehMcG85] Narain Gehani and Andrew McGettrick, editors.
Software Specification Techniques. Addison-Wesley, 1985.

[GogBur84a] J. A. Goguen and R. M. Burstall.
Introducing institutions.
In Edmund Clarke and Dexter Kozen, editors, *Logics of Programs, 1983*, volume 164 of *Lecture Notes in Computer Science*, pages 221–256. Springer-Verlag, 1984.
Extended version as CSLI report 85-30, Institutions: Abstract Model Theory for Computer Science.

[GogBur84b] J. A. Goguen and Rod M. Burstall.
Some fundamental algebraic tools for the semantics of computation, part 1.
Theoretical Computer Science, 31(1,2):175–209, 1984.

[GogBur84c] J. A. Goguen and Rod M. Burstall.
Some fundamental algebraic tools for the semantics of computation, part 2.
Theoretical Computer Science, 31(3):263–295, 1984.

[GogMes86] Joseph A. Goguen and J. Meseguer.
EQLOG: Equality, types, and generic modules for logic programming.
In D. DeGroot and G. Lindstrom, editors, *Logic Programming: Functions, relations, and equations*, pages 80–149. Prentice-Hall, 1986.

[GogMes87] Joseph A. Goguen and J. Meseguer.
Order-sorted algebra solves the constructor-selector, multiple representation and coercion problems.
In *2nd Symposium on Logic in Computer Science*, pages 18–29. IEEE Computer Society Press, 1987.

[GogWin88] Joseph Goguen and Timothy Winkler.
Introducing OBJ3.
Technical Report SRI-CSL-88-9, SRI International, August 1988.

[Gog$^+$78] Joseph Goguen, James Thatcher, and Eric Wagner.
An initial algebra approach to the specification, correctness and implementation of abstract data types.
In Raymond T. Yeh, editor, *Current Trends in Programming Methodology IV*, pages 80–149. Prentice-Hall, 1978.

[Gog$^+$85] Joseph A. Goguen, J.-P. Jouannaud, and J. Meseguer.
Operational semantics of order-sorted algebra.
In *Internatial Colloquium on Automata, Logic and Programming*, volume 194 of *Lecture Notes in Computer Science*, pages 221–231. Springer-Verlag, 1985.

[Gol84] Robert Goldblatt.
Topoi: The Categorial analysis of logic; Revised edition.
Number 98 in Studies in Logic and the Foundations of Mathematics. North Holland Publishing Company (Elsevier Science), 1984.

[Gra87] John W. Gray.
Categorical aspects of data type constructors.
Theoretical Computer Science, 50(2):103–135, 1987.

[Gra88] John W. Gray.
A categorical treatment of polymorphic operations.
In Main et al. [Mai$^+$88], pages 2–22.

[Gra89a] John W. Gray.
The category of sketches as a model for algebraic semantics.
In Gray and Scedrov [GraSce89], pages 109–135.

[Gra89b] John W. Gray.
The integration of logical and algebraic types.
In Ehrig et al. [Ehr$^+$89], pages 16–35.

[GraSce89] John W. Gray and Andre Scedrov, editors.
Categories in Computer Science and Logic, volume 92 of *Contemporary Mathematics*. American Mathematical Society, 1989.

[Gui82a] René Guitart.
From where do figurative algebras come?
Diagrammes, 7:RG1–RG7, 1982.

[Gui82b] René Guitart.
Getting started towards algebraic analysis.
Diagrammes, 7:G1–G3, 1982.

[Gui82c] René Guitart.
Qu'est-ce que la logique dans une catégorie?
Cahiers de Topologie et Géométrie Différentielle, XXIII(2):115–148, 1982.

[Gui86a] René Guitart.
Introduction à l'Analyse algébrique I.
Math. et Sc. Humaines, 24(96):49–63, 1986.

[Gui86b] René Guitart.
On the geometry of computations.
Cahiers de Topologie et Géométrie Différentielle, XXVII(4):107–136, 1986.

[Gui87] René Guitart.
Introduction à l'Analyse algébrique II.
Math. et Sc. Humaines, 98(?), 1987.

[Gui88] René Guitart.
On the geometry of computations II.
Cahiers de Topologie et Géométrie Différentielle, XXIX(4):297–326, 1988.

[GuiLai80] René Guitart and C. Lair.
Calcul syntaxique des modéles et calcul des formules internes.
Diagrammes, 4:1–106, 1980.

[GuiLai81a] René Guitart and C. Lair.
Esquisses de morphismes: Critères de rigidification des morphismes souples entre structures internes.
Diagrammes, 5:GL1–GL17, 1981.

[GuiLai81b] René Guitart and C. Lair.
Existence de diagrammes localement libres.
Diagrammes, 6:GL1–GL13, 1981.

[GuiLai82a] René Guitart and C. Lair.
Existence de diagrammes localement libres, II.
Diagrammes, 7:RGCL1–RGCL4, 1982.

[GuiLai82b] René Guitart and C. Lair.
Limites et colimites pour representer les formules.
Diagrammes, 7:GL1–GL24, 1982.

[Hag87a] Tatsuya Hagino.
A Categorical Programming Language.
PhD thesis, Department of Computer Science, University of Edinburgh, September 1987.

[Hag87b] Tatsuya Hagino.
A typed lambda calculus with categorical type constructors.
In David H. Pitt et al., editors, *Category Theory and Computer Science 1987*, volume 283 of *Lecture Notes in Computer Science*, pages 140–157. Springer-Verlag, 1987.

[Joh77] Peter T. Johnstone.
Topos Theory, volume 10 of *London Mathematical Society Monographs.*
Academic Press, Inc, 1977.

[Kea75] Orville Keane.
Abstract Horn theories.
In F. W. Lawvere, C. Maurer, and G. C. Wraith, editors, *Model Theory and Topoi*, volume 445 of *Lecture Notes in Mathematics*, pages 15–50. Springer-Verlag, 1975.

[Kel82a] G. M. Kelly.
Basic Concepts of Enriched Category Theory, volume 64 of *London Mathematical Society Lecture Note Series.*
Cambridge University Press, 1982.

[Kel82b] G. M. Kelly.
On the essentially-algebraic theory generated by a sketch.
Bulletin Australian Mathematical Society, 26(1):45–56, 1982.

[Kel82c] G. M. Kelly.
Structures defined by finite limits in the enriched context, I.
Cahiers de Topologie et Géométrie Différentielle, XXIII(1):3–42, 1982.

[KocRey77] A. Kock and G. E. Reyes.
Doctrines in categorical logic.
In Barwise [Bar77], pages 283–313.

[Lai70] C. Lair.
Constructions d'esquisses. Transformations naturelles généralisées.
Esquisses Mathématiques, 2, 1970.

[Lai74] C. Lair.
Dualité pour les structures algébriques esquissées.
Cahiers de Topologie et Géométrie Différentielle, XV(4):353–376, 1974.

[Lai75] C. Lair.
Etude générale de la catégorie des esquisses.
Esquisses Mathématiques, 23:1–62, 1975.

[Lai77] C. Lair.
Fermeture standard des catégories algébriques II.
Cahiers de Topologie et Géométrie Différentielle, XVIII(1):3–60, 1977.

[Lai79a] C. Lair.
Catégories modelables et catégories esquissables.
Diagrammes, 6:L1–L20, 1979.

[Lai79b] C. Lair.
Condition syntaxique de triplabilité d'un foncteur algébrique esquissé.
Diagrammes, 1:CL1–CL16, 1979.

[Lai87a] C. Lair.
Catégories qualifiables et catégoriques esquissables.
Diagrammes, 17, 1987.

[Lai87b] C. Lair.
Trames et semantiques catégoriques des systèmes de trames.
Diagrammes, 18, 1987.

[LamSco86] J. Lambek and P. J. Scott.
Introduction to higher order categorical logic, volume 7 of *Cambridge studies in advanced mathematics.*
Cambridge University Press, 1986.

[Law63] F. William Lawvere.
Functorial semantics of algebraic theories.
Proc. Nat. Acad. Sci., 50:869–872, 1963.

[Lin90] Olle Lindeberg.
Sketches and knowledge representation.
To appear, 1990.

[Mac80] Saunders MacLane.
The genesis of mathematical structures, as exemplified in the work of Charles Ehresmann.
Cahiers de Topologie et Géométrie Différentielle, XXI(4):353–365, 1980.

[Mac84] Saunders MacLane.
Diagrams, equations and theories in categories.
In P. Burmeister et al., editors, *Universal Algebra and its Links with Logic, Algebra, Combinatorics, and Computer Science*, pages 143–149. Heldermann Verlag, 1984.

[Mai+88] M. Main et al., editors.
Mathematical Foundations of Programming Language Semantics, 3rd workshop, volume 298 of *Lecture Notes in Computer Science.*
Springer-Verlag, 1988.

[Mak84] J. A. Makowsky.
Model theoretic issues in theoretical computer science, part I: Relational data bases and abstract data types.
In G. Lolli, G. Longo, and A. Marcja, editors, *Logic Colloquium '82*, volume 112 of *Studies in Logic and the Foundations of Mathematics*, pages 303–343. North Holland Publishing Company (Elsevier Science), 1984.

[Mak87] J. A. Makowsky.
Why Horn formulas matter in computer science: Initial structures and generic examples.
Journal of Computer and System Sciences, 34(2/3):266–292, 1987.

[MakPar89] Michael Makkai and Robert Paré.
Accessible Categories: The Foundations of Categorical Model Theory, volume 104 of *Contemporary Mathematics.*
American Mathematical Society, 1989.

[MakRey77] Michael Makkai and Gonzalo E. Reyes.
First Order Categorical Logic, volume 611 of *Lecture Notes in Mathematics.*
Springer-Verlag, 1977.

[Mat85] Philippe Mathieu.
Une méthode d'apprentissage des dépendances fonctionelles dans les bases de données déductives.
In *Proc. of the 1st Spanish Meeting on AI and Databases*, Blanes, October 1985.

[Mat86] Philippe Mathieu.
On the learning of functional dependencies in deductive databases.
Research Report SICS R86008, Swedish Institute of Computer Science, Box 1263, S-164 28 Kista, Sweden, 1986.
Extended abstract of [Mat85].

[Mou84] F. Mouen.
Caractérisation sémantique des catégories de structures.
Diagrammes, 11, 1984.

[Nai89] M. A. Nait Abdallah.
A logico-algebraic approach to the model theory of knowledge.
Theoretical Computer Science, 66:205–232, 1989.

[Pad88] Peter Padawitz.
Computing in Horn Clause Theories, volume 16 of *EATCS Monographs on Theoretical Computer Science.*
Springer-Verlag, 1988.

[Par89] Robert Paré.
Some applications of categorical model theory.
In Gray and Scedrov [GraSce89], pages 325–340.

[Per90] Oskar Permvall.
Categories, Sketches and Databases.
To appear, 1990.

[Pie88] Benjamin C. Pierce.
A taste of category theory for computer scientists.
Technical Report CMU-CS-88-203, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213, 1988.

[Rey74] Gonzalo E. Reyes.
From sheaves to logic.
In Aubert Daigneault, editor, *Studies in Algebraic Logic*, volume 9 of *Studies in Mathematics*, pages 143–204. The Mathematical Association of America, 1974.

[Rey77] Gonzalo E. Reyes.
Sheaves and concepts: A model-theoretic interpretation of Grotendieck topoi.
Cahiers de Topologie et Géométrie Différentielle, XVIII(2):105–137, 1977.

[Ser+90] Cristina Sernadas, José Fiadeiro, and Amílcar Sernadas.
Modular construction of logic knowledge bases: An algebraic approach.
Information Systems, 15(1):37–59, 1990.

[Sha88] Lokendra Shastri.
Semantic Networks: An Evidential Formalization and its Connectionist Realization.
Research Notes in Artificial Intelligence. Pitman, London, 1988.

[Sha89] Lokendra Shastri.
Default reasoning in semantic networks: A formalization of recognition and inheritance.
Journal of Artificial Intelligence, 39:283–355, 1989.

[Tha+82] J. W. Thatcher, E. G. Wagner, and J. B. Wright.
Data type specification: Parameterization and the power of specification techniques.
ACM Transactions on Programming Languages and Systems, 4(4):711–732, 1982.

[Wag90] E. G. Wagner.
Algebras, polynomials and programs.
Theoretical Computer Science, 70(1):3–34, 1990.

[Wel90] Charles Wells.
A generalization of the concept of sketch.
Theoretical Computer Science, 70(1):159–178, 1990.

[WelBar88] Charles Wells and Michael Barr.
The formal description of data types using sketches.
In Main et al. [Mai+88], pages 490–527.

[Wra89] G. C. Wraith.
A note on categorical datatypes.
In David H. Pitt et al., editors, *Category Theory and Computer Science 1989*, volume 389 of *Lecture Notes in Computer Science*, pages 118–127. Springer-Verlag, 1989.